Mechatronics 19 (2009) 695–704

Contents lists available at ScienceDirect

Mechatronics

journal homepage: www.elsevier.com/locate/mechatronics

Model based real time monitoring for collision detection of an industrial robot

Khaled Fawaz *, Rochdi Merzouki, Belkacem Ould-Bouamama

LAGIS, UMR CNRS 8146, Ecole Polytechnique de Lille, Avenue Paul Langevin, 59655 Villeneuve d'Ascq, France

ARTICLE INFO

Article history:
Received 16 July 2008
Accepted 4 February 2009

Keywords:
Fault Detection and Isolation (FDI)
Industrial robot
Dynamic modeling
Co-simulation
Bond graph

ABSTRACT

This paper deals with a model based real-time virtual simulator of industrial robot in order to detect eventual external collision. The implemented method concerns a model based Fault Detection and Isolation used to determine any lock of motion from an actuated robot joint after contact with static obstacles. Online implementation has been done in order to validate the proposed approach.

1. Introduction

The fast modernization of the production equipments makes that the industrial systems become increasingly complex and very sophisticated. In particular, the use of robots in modern manufacturing industries was intensively increased during these last decades. Robot technology has been applied in many aspects such as industry, agriculture, family service and medical etc. These robots carry out without slackening repetitive tasks. In the assembly lines of auto industry for example, they replace the workmen in the painful and dangerous tasks like (painting, welding, stamping, etc.). Massive use of the robots requires monitoring in continuous time to avoid any kind of collisions or abnormal operations which are likely cause considerable losses. Simulation software (RobotStudio [20], SimPro [21]) proposed by the manufacturers of the robots in order to help users to validate their programs offline without interruption are often very expensive, moreover, these tools does not make it possible to have a real-time monitoring on the operation of the real robots.

Many applications were developed in robotics during this last years in order to detect or to avoid collisions with an obstacle in the workspace of robot. Most of these applications are based on a virtual simulation. In [4] a robot simulation and monitoring system are described in order to establish a virtual environment using Java3D. It introduces the working principle and systematic architecture of a robot-oriented simulation and monitoring system based on Java3D and closed-loop feedback. In [7], a method simulating a redundant manipulator and the existing obstacle in their workspace with convex volumes, in order to avoid a collision is presented, while in [8] the inverse kinematic problem of a spatial redundant or non-redundant manipulator considering the criteria of collision avoidance and the joint functionality limits is solved.

In this paper, a 3D virtual simulator of an industrial robot with six degrees of freedom is developed. This simulator is based on the geometrical model of the manipulator robot and dynamical models of the six actuators which compose the joints of the robot. The main interest of this simulator is the real time monitoring of the real robot in real operation, allowing the detection of collisions.

For modeling the joint actuator system, one needs a unified approach as bond graph tool [3], to represent the multiphysical aspect of the electromechanical system and to exploit the structural and causality properties for generating the diagnostic algorithm [1,9].

The innovative contribution through this work concerns the use of the model based Fault Detection and Isolation (FDI) theory [5,10,12], in detecting and isolating collisions on industrial robot manipulator. This is done by using an online virtual simulator of the robot based on the robot and the joint actuators models, by informing the supervision system of any external collisions through the evaluation in time domain of the residuals.

This paper is organized as follows: after this section, the industrial robot system and online virtual simulator are presented in Section 2. Then, Section 3 regroups all the modeling of the studied system. The FDI algorithm is presented in Section 4 while in Section 5 the experimental results demonstrate the aim of the monitoring approach for industrial context.

2. System description and simulator architecture

The industrial studied system is a manipulator robot IRB140 of Fig. 1. This industrial robot is not accessible to the control

* Corresponding author. Tel.: +33 320436684; fax: +33 320337189.
E-mail address: khaled.fawaz@polytech-lille.fr (K. Fawaz).

0957-4158/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mechatronics.2009.02.005

(embedded control loop). This robot offers a single combination combining fast acceleration, important working area and high load capacity. It carries out six DOF (degrees of freedom) characterized by rotation motions. The first three articulations of this manipulator characterize for the first a rotation around a vertical axis, the second and the third following two horizontal axes whose motions are identified by the variables θs_1, θs_2 and θs_3, while the three last are characterized by the articular variables θs_4, θs_5 and θs_6.

2.1. Communication with the Robot

To communicate with the robot, one used the *WebWare SDK* [17]: software development kit, used to create PC-operator interfaces that communicate and interact with the ABB-robot controller through an Ethernet network. It allows:

- Transferring program between the PC and robot,
- Reading and writing robot I/O and RAPID variables,
- Starting and stopping robot execution,
- Receiving messages and events from the robot,
- Reading current robot position.

The WebWare SDK supports windows applications created with Microsoft Visual Basic. His interface consists of a set of OLE Custom Controls known as ActiveX controls. It includes four robot specified ActiveXs; one of them is the Helper control that provides methods, properties, and events to expose the entire S4 communication interface. The WebWare SDK is based on the ABB Interlink communication module, a core component for communication with the ABB-robot controller over the TCP/IP network protocol.

The WebWare SDK supports windows applications created with Microsoft Visual Basic, and the virtual 3D simulator is created with DevC++. To communicate with the two applications, we used the pipe which is a section of shared memory that processes use for communication. The process that creates a pipe is the pipe server. A process that connects to a pipe is a pipe client. One process writes information to the pipe, and then the other process reads the information from the pipe. In our case, application developed with visual basic is considered as the pipe server, while the application developed with DevC++ is the pipe client. Communication between the two applications is showed in Fig. 2.

2.2. 3D virtual simulator

The Virtual simulator was developed in C++ and includes a graphical representation based on OpenGL under Microsoft Windows (XP) operating systems (Fig. 3). Through the local area network connected with the robot, simulation system can receive the real-time data from the real system and dynamically display those data in a 3D graphics mode or reserve them for analysis. The virtual simulator proposed in this paper, is an assistant and supplementary tool to real robot system, helping operators to observe working condition from every view angles while being distant.

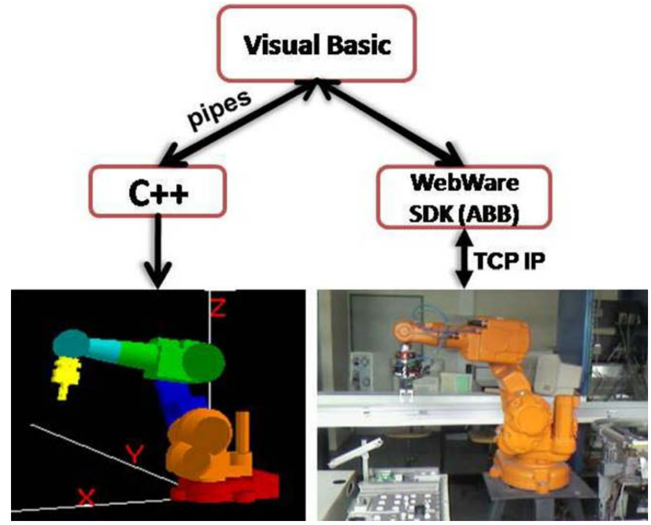

Fig. 2. Communication architecture.

3. System modeling

In this section, a description of the studied system parts models is introduced with the following scheduling:

1. Direct geometric model of the robot;
2. Inverse geometric model of the robot according to planned trajectory;
3. Direct dynamic model of the robot;
4. Dynamic models of joint actuators.

3.1. Direct geometrical model

Direct geometrical model is the whole relations which allow to calculate the position of the end-effectors, i.e. the operational coor-

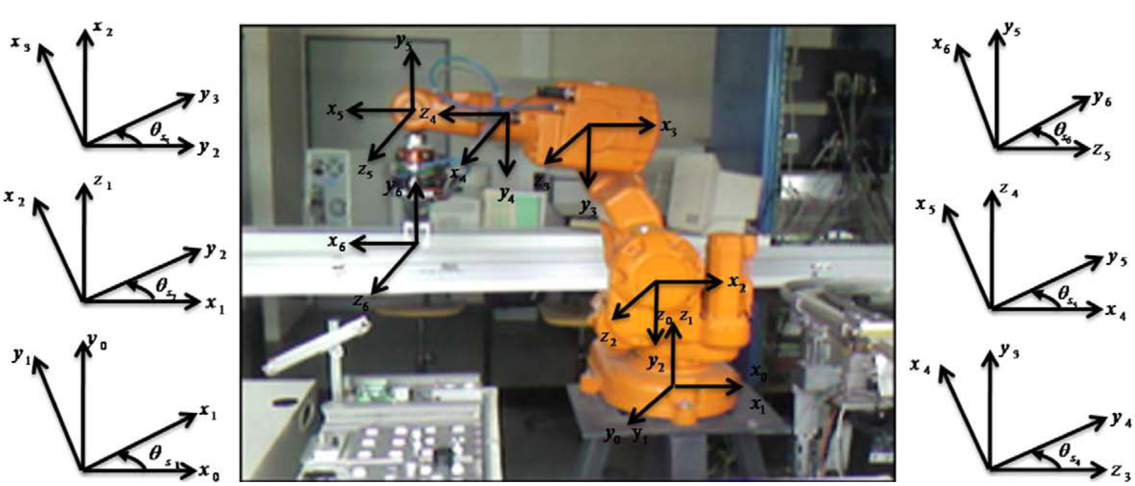

Fig. 1. Robot ABB-IRB 140.